

AMENDMENTS TO THE CLAIMS

1. (Previously Presented) A method, comprising:
 - loading a bootstrap program into an area of a memory of a service module that was occupied by a run time program, the bootstrap program loaded after the service module is reset due to an error while executing the run time program;
 - capturing a memory image of the memory using the bootstrap program; and
 - sending the captured memory image to a control module using a bus shared by the control module and the service module, wherein the control module is configured to receive captured memory images from one or more service modules.
2. (Currently Amended) The method of claim 1, further comprising allocating communication buffers used by the bootstrap program, to transfer the captured memory image to the control module, in the area of memory that ~~was~~is occupied by the run time program.
3. (Currently Amended) The method of claim ~~1~~2, wherein the captured memory image of the memory is compressed before being sent to the control module.
4. (Original) The method of claim 1, wherein capturing the memory image comprises:
 - reading a first block of memory; and
 - compressing the first block of memory into a compressed unit before reading a second block of memory using a compression algorithm.
5. (Original) The method of claim 4, wherein one or more blocks of memory is compressed into the compressed unit until the compressed unit reaches a predetermined size.
6. (Currently Amended) The method of claim ~~5~~3, wherein sending the captured memory image to the control module comprises sending one or more compressed units of the captured memory image to the control module without waiting for the entire memory image to be compressed.

7. (Previously Presented) The method of claim 6, wherein the one or more compressed units is stored as a file in a persistent storage of the control module.
8. (Original) The method of claim 4, wherein the compression algorithm is a zip algorithm.
9. (Original) The method of claim 1, further comprising loading the run time program into the memory of the service module after sending the captured memory image to the control module.
10. (Currently Amended) A computer readable medium having stored thereon sequences of instructions which are executable by a system, and which, when executed by the system, cause the system to:
 - load a bootstrap program into an area of a memory of a service module ~~that was occupied by a run time program~~ thereby overwriting a code section, and not a data section, of the run time program, the bootstrap program loaded after the service module is reset due to an error while executing the run time program, wherein the service module does not have persistent storage capability for a core dump;
 - capture a memory image of the memory using the bootstrap program; and
 - send the captured memory image to a control module using a bus shared by the control module and the service module, wherein the control module is configured to receive captured memory images from one or more service modules.
11. (Currently Amended) The computer readable medium of claim 10, further comprising instructions to allocate communication buffers used by the bootstrap program, to transfer the captured memory image to the control module, in the area of memory ~~that was occupied by~~ the code section, not the data section, of the run time program.
12. (Original) The computer readable medium of claim 10, wherein the captured memory image of the memory is compressed before being sent to the control module.
13. (Currently Amended) The computer readable medium of claim 10 further comprising instructions to examine a reset reason that was stored in the system (a) before the service module is reset, and (b) when an error occurs in the service module,

~~to see if a core dump is required, wherein the instructions to capture the memory image comprises instructions to:~~

~~—— read a first block of memory; and~~

~~—— compress the first block of memory into a compressed unit before reading a second block of memory using a compression algorithm.~~

14. (Currently Amended) The computer readable medium of claim 13, wherein one or more blocks of memory is compressed into ~~the~~ a compressed unit until the compressed unit reaches a predetermined size.

15. (Currently Amended) The computer readable medium of claim 14, wherein the instructions to send the captured memory image to the control module comprises instructions to send one or more compressed units to the control module without waiting for the entire memory image to be compressed.

16. (Currently Amended) The computer readable medium of claim 15, wherein the one or more compressed units is stored as a file in a persistent storage of the control ~~unit~~ module.

17. (Original) The computer readable medium of claim 13, wherein the compression algorithm is a zip algorithm.

18. (Original) The computer readable medium of claim 10, further comprising instructions to load the run time program into the memory of the service module after sending the captured memory image to the control module.

19. (Currently Amended) A system, comprising:
a memory; and

a processor coupled with the memory, the processor configured by a bootstrap program to capture a memory image of the memory ~~after in response to~~ the processor ~~is being~~ reset when an error occurs while executing a run time program, wherein the bootstrap program is loaded into an area of the memory occupied by the run time program when the error occurs.

20. (Original) The system of claim 19, wherein the captured memory image is sent out to a bus using communication buffers allocated in the area of the memory occupied by the run time program, the memory areas occupied by the bootstrap program and allocated to the communication buffers do not overlap.
21. (Original) The system of claim 20, wherein the captured memory image is sent out to the bus in compressed form.
22. (Currently Amended) A system comprising:
a memory means;
means for loading a bootstrap program into a first memory area of the memory means, ~~the memory area previously~~ that is occupied by a run time program thereby overwriting a code portion, not a data portion, of the run time program, the bootstrap program loaded after a reset due to an error while executing the run time program;
means for capturing a memory image of the memory means; and
means for transferring the captured memory image to a control module.
23. (Original) The system of claim 22, wherein the means for capturing the memory image comprises means for capturing an image of a second memory area used by the run time program as a data area when the error occurs.
24. (Original) The system of claim 22, wherein the means for transferring the captured memory image comprises means for compressing the captured memory image.
25. (Original) The system of claim 24, wherein the captured memory image is transferred using communication buffers allocated in the first memory area.